ACCES I/O PRODUCTS INC
10623 Roselle Street, San Diego, CA 92121
TEL (858)550-9559 FAX (858)550-7322

MODEL PCI-COM422/8

USER MANUAL

FILE: MPCICOM422-8.D1d

Notice

The information in this document is provided for reference only. ACCES does not assume any liability arising out of the application or use of the information or products described herein. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of ACCES, nor the rights of others.

IBM PC, PC/XT, and PC/AT are registered trademarks of the International Business Machines Corporation.

Printed in USA. Copyright 1995 by ACCES I/O Products Inc, 10623 Roselle Street, San Diego, CA 92121. All rights reserved.

Warranty

Prior to shipment, ACCES equipment is thoroughly inspected and tested to applicable specifications. However, should equipment failure occur, ACCES assures its customers that prompt service and support will be available. All equipment originally manufactured by ACCES which is found to be defective will be repaired or replaced subject to the following considerations.

Terms and Conditions

If a unit is suspected of failure, contact ACCES' Customer Service department. Be prepared to give the unit model number, serial number, and a description of the failure symptom(s). We may suggest some simple tests to confirm the failure. We will assign a Return Material Authorization (RMA) number which must appear on the outer label of the return package. All units/components should be properly packed for handling and returned with freight prepaid to the ACCES designated Service Center, and will be returned to the customer's/user's site freight prepaid and invoiced.

Coverage

First Three Years: Returned unit/part will be repaired and/or replaced at ACCES option with no charge for labor or parts not excluded by warranty. Warranty commences with equipment shipment.

Following Years: Throughout your equipment's lifetime, ACCES stands ready to provide on-site or in-plant service at reasonable rates similar to those of other manufacturers in the industry.

Equipment Not Manufactured by ACCES

Equipment provided but not manufactured by ACCES is warranted and will be repaired according to the terms and conditions of the respective equipment manufacturer's warranty.

General

Under this Warranty, liability of ACCES is limited to replacing, repairing or issuing credit (at ACCES discretion) for any products which are proved to be defective during the warranty period. In no case is ACCES liable for consequential or special damage arriving from use or misuse of our product. The customer is responsible for all charges caused by modifications or additions to ACCES equipment not approved in writing by ACCES or, if in ACCES opinion the equipment has been subjected to abnormal use. "Abnormal use" for purposes of this warranty is defined as any use to which the equipment is exposed other than that use specified or intended as evidenced by purchase or sales representation. Other than the above, no other warranty, expressed or implied, shall apply to any and all such equipment furnished or sold by ACCES.

Table of Contents

Chapter 1: Introduction 1-1
RS422 Balanced Mode Operation 1-1
COM Port Compatibility 1-1
Communication Mode 1-1
Baud Rate Ranges 1-2
Specifications 1-3
Chapter 2: Installation 2-1
CD Installation 2-1
3.5-Inch Diskette Installation 2-1
Directories Created on the Hard Disk 2-2
Installing the Card 2-4
Windows NT 4.0 Installation Instructions for the PCI-COM422/8 2-5
Chapter 3: Option Selection 3-1
Chapter 4: Address Selection 4-1
Chapter 5: Programming 5-1
Sample Programs 5-1
Windows Programming 5-1
Initialization 5-1
Reception 5-3
Transmission 5-4
Chapter 6: Connector Pin Assignments 6-1
Input/Output Connections 6-1

List of Figures

Figure 1-1: PCI-COM422/8 Block Diagram . . . . . Page 1-4
Figure 3-1: Simplified Termination Schematic . . . . . Page 3-1
Figure 3-2: PCI-COM422/8 Option Selection Map . . . . . Page 3-2
Figure 3-3: Low-Profile PCI-COM485/8 Option Selection Map . . . . . Page 3-3

List of Tables

Table 5-1: Baud Rate Divisor Values . . . . . Page 5-2
Table 6-1: Connection Pin Assignments . . . . . Page 6-1
Table 6-2: Data Cable Wiring . . . . . Page 6-2

Chapter 1: Introduction

The PCI-COM422/8 Serial Interface Card was designed for effective transmission in RS422 (EIS422) protocol. The card is 7.83 inches long and may be installed in PCI-bus slots of IBM PC or compatible computers. The card features eight independent, asynchronous RS422 serial ports, type 16550 buffered UARTs.

RS422 Balanced Mode Operation

The PCI-COM422/8 supports RS422 communications and uses differential balanced drivers for long range and noise immunity. PCI-COM422/8 also has the capability to add load resistors to terminate the communications lines. RS422 communications requires that a transmitter supply a bias voltage to ensure a known "zero" state. Also, receiver inputs at each end of the network should be terminated to eliminate "ringing". The PCI-COM422/8 supports biasing by default and supports termination by jumpers on the card. If your application requires the transmitter to be un-biased, please contact the factory.

COM Port Compatibility

Type 16550 UARTs are used as the Asynchronous Communication Element (ACE). These include a 16-byte transmit/receive buffer to protect against lost data in multitasking operating systems, while maintaining 100% compatibility with the original IBM serial port. The system assigns the address(es).

A crystal oscillator is located on the card. This oscillator permits precise selection of baud rates up to 115,200 or, by changing a jumper, up to 921,600 with the standard crystal oscillator.

The driver/receiver used, the SN75176B, is capable of driving extremely long communication lines at high baud rates. It can drive up to ±60 mA on balanced lines and receive inputs as low as 200 mV differential signal superimposed on common mode noise of +12 V or -7 V. In case of communication conflict, the driver/receivers feature thermal shutdown.

Communication Mode

PCI-COM422/8 supports Full-Duplex and Half-Duplex communications with a 4-wire cable connection. Half-Duplex allows traffic to travel in both directions, but only one way at a time.

Baud Rate Ranges

The card has capability for two baud rate range selections and can be selected through the setting of the CLK X1/CLK X4 jumper. The baud rates are selected on a port-by-port basis. One range is for up to 115,200 baud applications and the other is up to 921,600 baud applications. The lower baud rate is selected when the jumper is in the CLK X1 position. When the jumper is in the X1 position we can utilize the industry standard baud rate selection methodology. When in the X4 position, we must compensate for the new input frequency by letting the Windows operating system see the frequency at 1/4 of the selected value.

As an example when the X4 position is selected, assume the selected baud output is to be 9600 baud. The operating system would need to have defined the value of 2400 baud so that the correct system operations occur.

Note: Refer to the Baud Rate Divisor Value Table on page 5-1 of the manual.

Specifications

Communications Interface

- I/O Connection: 50-pin SCSI D-Connector
- Serial Ports: Eight cable terminated shielded male D-sub 9-pin with standard IBM AT Style connectors compatible with RS422 specifications.
- Character length: 5, 6, 7, or 8 bits.
- Parity: Even, odd or none.
- Stop Interval: 1, 1.5, or 2 bits.
- Serial Data Rates: Up to 115,200 baud, Asynchronous, A faster range of rates, up to 921,600, is achieved by jumper selection on the card. Type 16550 buffered UART.
- Address: Continuously mappable within 0000 to FFFF (hex) range of PCI bus addresses.
- Receiver Input Sensitivity: ±200 mV, differential input.
- Common Mode Rejection: +12V to -7V
- Transmitter Output Drive Capability: 60 mA, with thermal shutdown.

Environmental

- Operating Temperature Range: 0 °C. to +60 °C.
- Storage temperature Range: -50 °C. to +120 °C.
- Humidity: 5% to 95%, non-condensing.

- Power Required: +5VDC at 125 mA typical, -12VDC at 5 mA typical, +12VDC at 5 mA typical, 750 mW total power consumption.
- Size: 7.8 Inches long (198 mm) by 3.8 inches high (97 mm).

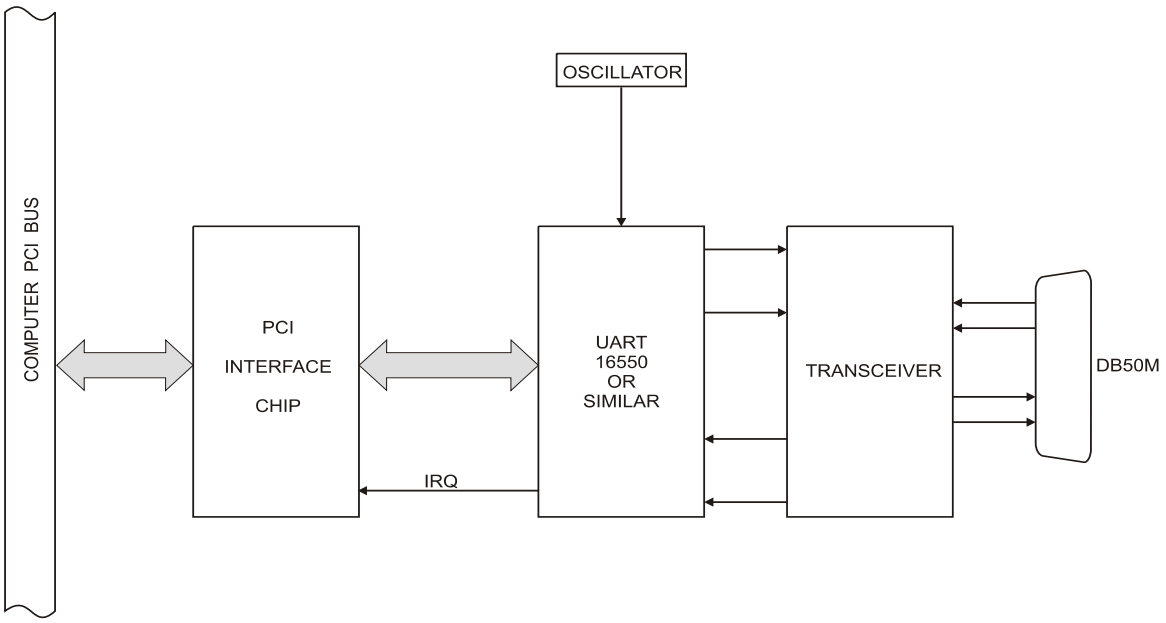

Figure 1-1: PCI-COM422/8 Block Diagram
(Only one serial channel shown)

Chapter 2: Installation

The software provided with this card is contained on either one CD or multiple diskettes and must be installed onto your hard disk prior to use. To do this, perform the following steps as appropriate for your software format and operating system. Substitute the appropriate drive letter for your CD-ROM or disk drive where you see d: or a: respectively in the examples below.

CD Installation

DOS/WIN3.x

1. Place the CD into your CD-ROM drive.
2. Type D : Enter to change the active drive to the CD-ROM drive.
3. Type I N S T A L L Enter to run the install program.
4. Follow the on-screen prompts to install the software for this card.

WIN95/98/NT

1. Place the CD into your CD-ROM drive.
2. The CD should automatically run the install program after 30 seconds. If the install program does not run, click START | RUN and type d:install, click OK or press Enter.
3. Follow the on-screen prompts to install the software for this card.

3.5-Inch Diskette Installation

As with any software package, you should make backup copies for everyday use and store your original master diskettes in a safe location. The easiest way to make a backup copy is to use the DOS DISKCOPY utility.

In a single-drive system, the command is:

D I S K C O P Y A : A : Enter

You will need to swap disks as requested by the system.
In a two-disk system, the command is:

D I S K C O P Y A : B : Enter

This will copy the contents of the master disk in drive A to the backup disk in drive B.

To copy the files on the master diskette to your hard disk, perform the following steps.

1. Place the master diskette into a floppy drive.
2. Change the active drive to the drive that has the diskette installed. For example, if the diskette is in drive A, type A : Enter.
3. Type I N S T A L L Enter and follow the on-screen prompts.

Directories Created on the Hard Disk

The installation process will create several directories on your hard disk. If you accept the installation defaults, the following structure will exist.

[CARDNAME]

Root or base directory containing the SETUP.EXE setup program used to help you configure jumpers and calibrate the card.

DOS\PSAMPLES: A subdirectory of [CARDNAME] that contains Pascal samples.
DOS\CSAMPLES: A subdirectory of [CARDNAME] that contains "C" samples.
Win32\language: Subdirectories containing samples for Win95/98 and NT.

WinRisc.exe

A Windows dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem.

ACCES32

This directory contains the Windows 95/98/NT driver used to provide access to the hardware registers when writing 32-bit Windows software. Several samples are provided in a variety of languages to demonstrate how to use this driver. The DLL provides four functions (InPortB, OutPortB, InPort, and OutPort) to access the hardware.

This directory also contains the device driver for Windows NT, ACCESNT.SYS. This device driver provides register-level hardware access in Windows NT. Two methods of using the driver are available, through ACCES32.DLL (recommended) and through the DeviceIOControl handles provided by ACCESNT.SYS (slightly faster).

SAMPLES

Samples for using ACCES32.DLL are provided in this directory. Using this DLL not only makes the hardware programming easier (MUCH easier), but also one source file can be used for both Windows 95/98 and WindowsNT. One executable can run under both operating systems and still have full access to the hardware registers. The DLL is used exactly like any other DLL, so it is compatible with any language capable of using 32-bit DLLs. Consult the manuals provided with your language's compiler for information on using DLLs in your specific environment.

VBACCES

This directory contains sixteen-bit DLL drivers for use with VisualBASIC 3.0 and Windows 3.1 only. These drivers provide four functions, similar to the ACCES32.DLL. However, this DLL is only compatible with 16-bit executables. Migration from 16-bit to 32-bit is simplified because of the similarity between VBACCES and ACCES32.

PCI

This directory contains PCI-bus specific programs and information. If you are not using a PCI card, this directory will not be installed.

SOURCE

A utility program is provided with source code you can use to determine allocated resources at run-time from your own programs in DOS.

PCIFind.exe

A utility for DOS and Windows to determine what base addresses and IRQs are allocated to installed PCI cards. This program runs two versions, depending on the operating system. Windows 95/98/NT displays a GUI interface, and modifies the registry. When run from DOS or Windows3.x, a text interface is used. For information about the format of the registry key, consult the card-specific samples provided with the hardware. In Windows NT, NTioPCI.SYS runs each time the computer is booted, thereby refreshing the registry as PCI hardware is added or removed. In Windows 95/98/NT PCIFind.EXE places itself in the boot-sequence of the OS to refresh the registry on each power-up.

This program also provides some COM configuration when used with PCI COM ports. Specifically, it will configure compatible COM cards for IRQ sharing and multiple port issues.

WIN32IRQ

This directory provides a generic interface for IRQ handling in Windows 95/98/NT. Source code is provided for the driver, greatly simplifying the creation of custom drivers for specific needs. Samples are provided to demonstrate the use of the generic driver. Note that the use of IRQs in near-real-time data acquisition programs requires multi-threaded application programming techniques and must be considered an intermediate to advanced programming topic. Delphi, C++ Builder, and Visual C++ samples are provided.

Findbase.exe

DOS utility to determine an available base address for ISA bus , non-Plug-n-Play cards. Run this program once, before the hardware is installed in the computer, to determine an available address to give the card. Once the address has been determined, run the setup program provided with the hardware to see instructions on setting the address switch and various option selections.

Poly.exe

A generic utility to convert a table of data into an nth order polynomial. Useful for calculating linearization polynomial coefficients for thermocouples and other non-linear sensors.

Risc.bat

A batch file demonstrating the command line parameters of RISCTerm.exe.

RISCTerm.exe

A dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem. RISCTerm stands for Really Incredibly Simple Communications TERMinal.

Installing the Card

The PCI-COM422/8 card can be installed in a five-volt PCI slot of an IBM or compatible computer. Before , carefully read the Option Selection section of this manual and configure the card according to your requirements. Finally, our SETUP.EXE program will lead you through the process of setting the options on the PCI-COM422/8. The setup program does not set the options. These must be set manually by jumpers on the card.

To Install the Card

1. Turn OFF computer power.
2. Remove the computer cover.
3. Install jumpers from either the Option Selection section of this manual or the suggestions of our SETUP.EXE software program.
4. Install the card in an available PCI-bus slot.
5. Replace the computer cover and turn the computer ON.
6. Enter the CMOS setup program of your system and verify that the PCI plug-and-play option is set appropriately for your system. Systems running Windows95 (or any other PNP-compliant Operating System) should set the CMOS option to OS. Systems running under DOS, WindowsNT 3.51, Windows 3.1, or any other non-PNP-compliant Operating System should set the PNP CMOS option to BIOS or Motherboard. Save the option and continue booting the system.

A "spider" cable is provided to interface between the 50-pin SCSI connector on the card, and your system cabling. Eight individual 9-pin connectors are provided.

If you are using Windows95, your operating system should detect the new hardware and prompt you for the installation disk. Insert the ACCES-provided diskette into the A drive and allow the operating system to look for the files that it wants.

The base address assigned by BIOS or the operating system can change each time new hardware is installed into or removed from the computer.

PCI cards are assigned resources by the system. To determine the resource assigned, run PCIFind or look in the operating systems list of Detected Devices. In Win95/98/NT, this is the Device Manager of the system applet, found in the control panel. See the Tools diskette for more information on PCIFind.

Windows NT 4.0 Installation Instructions for the PCI-COM422/8

Unlike Windows95, NT4.0 is not a plug-and-play compliant Operating System. Windows NT will not auto-detect and install new hardware devices. Therefore, in order to tell Windows NT that you have installed eight new COM ports, you must use the utilities provided by NT in the Control Panel.

Once you have installed the card in the computer and the system is booted to Windows NT 4.0 (Service Pack 3 or higher is recommended), you will also need to install several drivers and utilities provided. These utilities are designed to detect the location of the hardware and report the base address and IRQ assignments for your use. The TOOLS diskette contains these programs.

Place the TOOLS disk in a floppy drive (ex: A:), click the Start Menu, select the Run menu item, and type A:INSTALL [enter]. This will run the installation program from the floppy disk to install the software to your hard-disk.

Once the software installation is complete, run PCIFind.EXE from the root directory created on the hard disk. This program copies NTIOPCI.SYS to the [NT]\SYSTEM32\DRIVERS directory and dynamically loads and runs the driver.

Note

If you previously installed an older version of NTIOPCI.SYS, please delete the file from your [NT]\SYSTEM32\DRIVERS directory prior to running PCIFind.

PCIFind displays the list of COM addresses and the IRQ assigned and adds all pertinent registry entries. The following steps are only necessary if there is a problem running PCIFind and the registry is not updated.

Open the CONTROL PANEL (START|RUN|CONTROL, START|SETTINGS|CONTROL PANEL, or MY COMPUTER|CONTROL PANEL) and execute the PORTS applet.

Click Add...; a dialog box titled "Advanced Settings for New Port" will appear. Select the COM port number of an available COM port, type the Base I/O Port Address of COMA and the IRQ from the PCIFind screen. Click OK.

You have now installed one port in NT. Continue Clicking Add... and selecting addresses for the remainder of the eight ports.

All these steps are performed manually, but could be automated, by making registry entries in various registry keys. Consult your Microsoft provided documentation for more information.

The base addresses and IRQ of the PCI-COM422/8 is provided in the registry under the NTIOPCI key, in a structure of type PCI_COMMON_CONFIG. See the sample program for a demonstration of how to read this structure and extract the addresses in an application program.

Interrupts

Please note that, in Windows NT, changes must be made to the system registry to support IRQ sharing. The following is excerpted from "Controlling Multiport Serial I/O Cards" provided by Microsoft in the MSDN library, documentid:mk:@ivt:nt40res/D15/S55FC.HTM, also available in the WindowsNT Resource Kit.

The Microsoft serial driver can be used to control many dumb multiport serial cards. Dumb indicates that the control includes no on-board processor. Each port of a multiport card has a separate subkey under the HKLM\CurrentControlSet\Services\Serial\Parameters subkey in the registry. In each of these subkeys, you must add values for DosDevices, Interrupt, InterruptStatus, PortAddress, and PortIndex because these are not detected by the Hardware Recognizer. (For descriptions and ranges for these values, see Regentry.hlp, the Registry help file on the WindowsNT Workstation Resource Kit CD.)

For example, if you have an eight-port PCI-COM422/8 card configured to use address 0xFC00 with an interrupt of 05, the values in the Registry are:

Serial##### Subkey:
PortAddress = REG_DWORD 0xFC00
Interrupt = REG_WORD 5
DosDevices = REG_SZ COM5
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD 1

Serial##### Subkey:
PortAddress = REG_DWORD 0xFC20
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM9
InterruptStatus = REG_DWORD 0xFC40
PortIndex - REG_DWORD 5

Serial##### Subkey:
PortAddress = REG_DWORD 0xFC08
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM6
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD 2

Serial##### Subkey:
PortAddress = REG_DWORD 0xFC28
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM10
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD 6

Serial##### Subkey:
PortAddress = _DWORD 0xFC10
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM7
InterruptStatus = REG_DWORD 0xFC40
PortIndex - REG_DWORD 3

Serial##### Subkey:
PortAddress = REG_DWORD 0xFC30
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM11
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD 7

Serial##### Subkey:
PortAddress = REG_DWORD 0xFC18
Interrupt = REG_DWORD 5
Dos Devices = REG_SZ COM8
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD4

Serial##### Subkey:
PortAddress = REG_DWORD 0xFC38
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM12
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD8

As this example shows, the shared IRQ status Register is located at Base address + 0x40.

Chapter 3: Option Selection

To help you locate the jumpers described in this section, refer to the Option Selection Map at the end of this section. Operation of the serial communications section is determined by jumper installation as described in the following paragraphs.

Terminations

A transmission line should be terminated at the receiving end in its characteristic impedance. Installing a jumper at the locations labeled LDxO and LDxI apply a 120Ω load across the transmit/receive input/output for RS422 operation.

In RS422 operations where there are multiple terminals, only the RS422 ports at the receiver end of the network should have terminating impedance as described above. To so terminate the COM A port, place a jumper at the location labeled LDAO. To terminate the COM B, COM C, COM D, COM E, COM F and COM H ports, place jumpers at locations labeled LDBO, LDCO, LDDO, LDEO, LDFO LDGO and LDHO respectively. Similarly, Inputs are terminated at jumpers labeled LDAI, LDB1, etc.

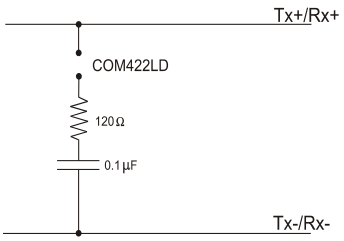

Figure 3-1: Simplified Termination Schematic

Also, for RS422 operation, there must be a bias on the RX+ and RX- lines. If the COM422/8 card is not to provide that bias, contact the factory technical support.

Data Cable Wiring

Signal	Pin Connection
Ain-	1
Aout+	2
Aout-	3
100 Ω to Ground	5
Ain+	9

Baud Rate Ranges

The jumpers labeled CLK X1 and CLK X4 provide means to select baud rates in either of two ranges. When in the "X1" position, the baud rate range is up to 115,200 baud. When in the CLK X4 position, the baud rate range is 200 to 921,600 baud.

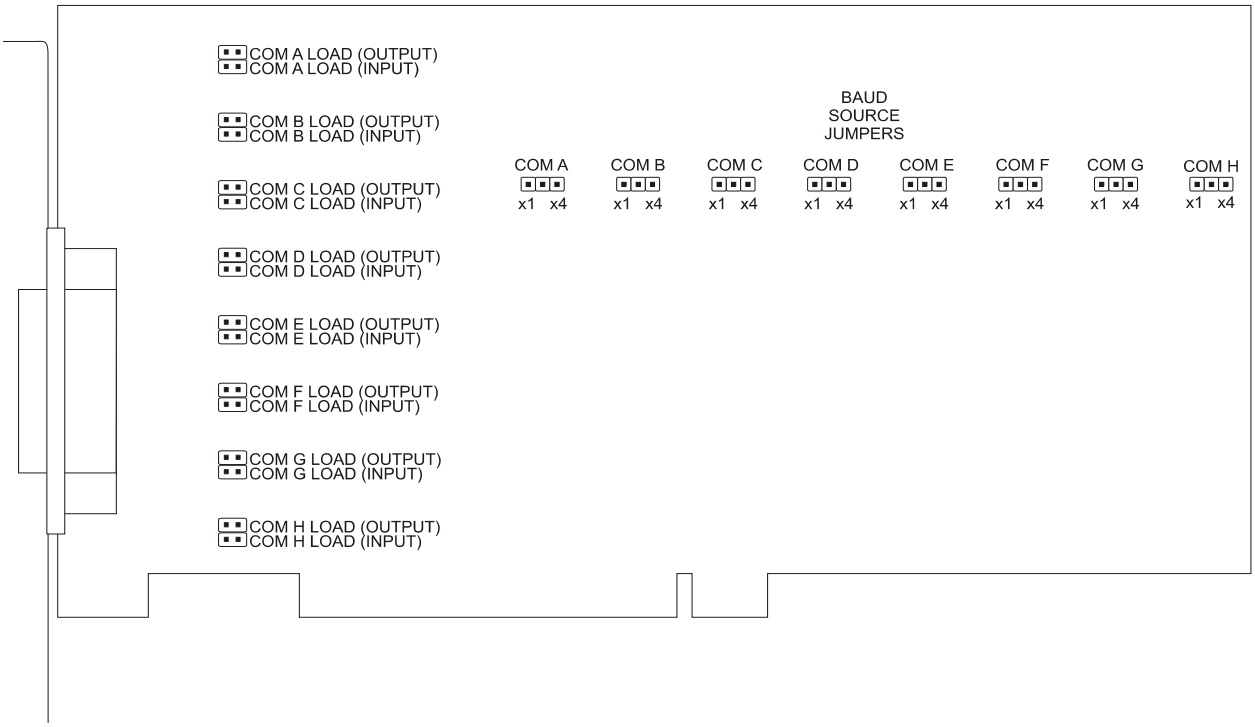

Figure 3-2: PCI-COM422/8 Option Selection Map

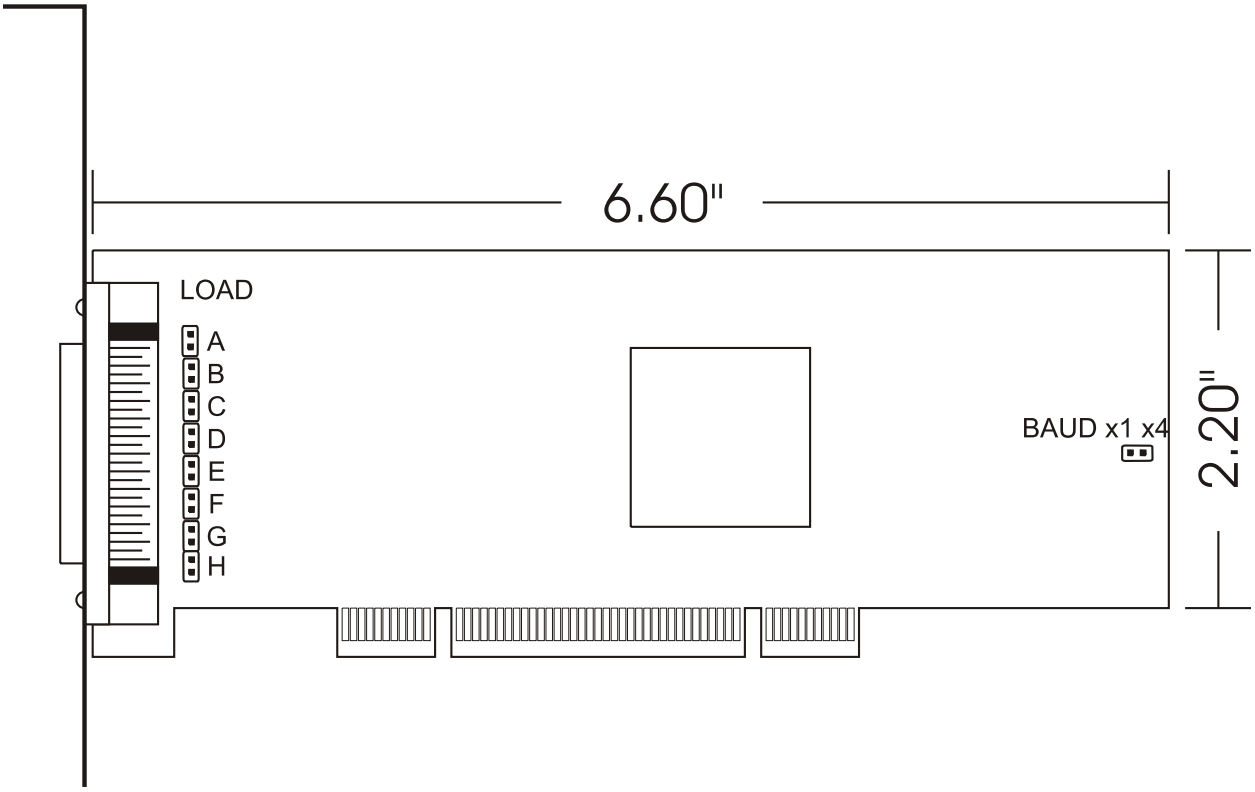

Figure 3-3: Low-Profile PCI-COM485/8 Option Selection Map

Chapter 4: Address Selection

The PCI-COM422/8 card uses one address space. COM A, COM B, COM C, COM D, COM E, COM F, COM G, and COM H each occupy eight consecutive register locations. The interrupt register which indicates which port or ports caused the interrupt is located at base address + 64.

PCI architecture is Plug-and-Play. This means that the BIOS or Operating System determines the resources assigned to PCI cards rather than you selecting those resources with switches or jumpers. As a result, you cannot set or change the card's base address. You can only determine what the system has assigned.

To determine the base address that has been assigned, run the PCIFind.EXE, or PCINT utility program provided. This utility will display a list of all of the ACCES cards detected on the PCI bus, the addresses assigned to each function on each of the cards, and the respective IRQs (if any) allotted.

Alternatively, some operating systems (Windows95/98/2000) can be queried to determine which resources were assigned. In these operating systems, you can use either PCIFind (DOS) or PCINT (Windows95/98/NT), or the Device Manager utility from the System Applet of the control panel. The PCI-COM422/8 is installed in the Data Acquisition class of the Device Manager list. Selecting the card, clicking Properties, and then selecting the Resources Tab will display a list of the resources allocated to the card.

The PCI bus supports 64K of I/O space, so your card's addresses may be located anywhere in the 0000 to FFFF hex range.

Vendor ID code is 494F (ASCII for "I/O")
Device ID code is 1068

Chapter 5: Programming

Sample Programs

There are sample programs provided with the PCI-COM422/8 card in C, Pascal, QuickBASIC, and several Windows languages. DOS samples are located in the DOS directory and Windows samples are located in the WIN32 directory.

Windows Programming

The PCI-COM422/8 card installs into Windows as COM ports. Thus the Windows standard API functions can be used. In particular:

- CreateFile() and CloseHandle() for opening and closing a port.
- SetupComm(), SetCommTimeouts(), GetCommState(), and SetCommState() to set and change a port's settings.
- ReadFile() and WriteFile() for accessing a port.

See the documentation for your chosen language for details.

Under DOS, the process is very different. The remainder of this chapter describes DOS programming.

Initialization

Initializing the chip requires knowledge of the UART's register set. The first step is to set the baud rate divisor. You do this by first setting the DLAB (Divisor Latch Access Bit) high. This bit is Bit 7 at Base Address +3. In C code, the call would be:

```
outportb(BASEADDR +3,0x80);
```

You then load the divisor into Base Address +0 (low byte) and Base Address +1 (high byte). The following equation defines the relationship between baud rate and divisor:

desired baud rate = (UART clock frequency) / (32 * divisor)

On the PCI-COM422/8 card, the UART clock frequency is 1.8432 MHz. On the next page is a table for the popular divisor frequencies.

Baud Rate	Divisor x1	Divisor x8	Max Diff. Cable Length *
921600	N/A	1	250 ft
460800	N/A	2	500 ft
230400	N/A	4	800 ft
153600	N/A	6	1300 ft
115200	1	8	2200 ft
57600	2	16	4000 ft
38400	3	24	4000 ft
28800	4	32	4000 ft
19200	6	48	4000 ft
14400	8	64	4000 ft
9600	12	96	4000 ft
4800	24	192	4000 ft
2400	48	384	4000 ft
1200	96	768	4000 ft

*These are theoretical maximums based on typical conditions and good quality cables based on the EIA 485 and EIA 422 standard for balanced differential drivers. RS232 communication lines have a maximum length of 50 feet, regardless of speed.

Table 5-1: Baud Rate Divisor Values

In C, the code to set the chip to 9600 baud is:

```
outportb(BASEADDR, 0x0C);
outportb(BASEADDR +1,0);
```

The second initializing step is to set the Line Control Register at Base Address +3. This register defines word length, stop bits, parity, and the DLAB.

Bits 0 and 1 control word length and allow word lengths from 5 to 8 bits. Bit settings are extracted by subtracting 5 from the desired word length.

Bit 2 determines the number of stop bits. There can be either one or two stop bits. If Bit 2 is set to 0, there will be one stop bit. If Bit 2 is set to 1, there will be two stop bits.

Bits 3 through 6 control parity and break enable. They are not commonly used for communications and should be set to zeroes.

Bit 7 is the DLAB discussed earlier. It must be set to zero after the divisor is loaded or else there will be no communications.

The C command to set the UART for an 8-bit word, no parity, and one stop bit is:

```
outportb(BASEADDR +3, 0x03)
```

Reception

Reception can be handled in two ways: polling and interrupt-driven. When polling, reception is accomplished by constantly reading the Line Status Register at Base Address +5. Bit 0 of this register is set high whenever data are ready to be read from the chip. A simple polling loop must continuously check this bit and read in data as it becomes available. The following code fragment implements a polling loop and uses a value of 13, (ASCII Carriage Return) as an end-of-transmission marker:

```
do
{
 while (!(inportb(BASEADDR +5) & 1)); /*Wait until data ready*/
   data[i++]= inportb(BASEADDR);
}
 while (data[i]!=13);  /*Reads the line until null character rec'd*/
```

Interrupt-driven communications should be used whenever possible and is required for high data rates. Writing an interrupt-driven receiver is not much more complex than writing a polled receiver but care should be taken when installing or removing your interrupt handler to avoid writing the wrong interrupt, disabling the wrong interrupt, or turning interrupts off for too long a period.

The handler would first read the Interrupt Identification Register at Base Address +2. If the interrupt is for Received Data Available, the handler then reads the data. If no interrupt is pending, control exits the routine. A sample handler, written in C, is as follows:

```
readback = inportb(BASEADDR +2);
if (readback & 4)        /*Readback will be set to 4 if data are available*/
data[i++]=inportb(BASEADDR);
outportb(0x20,0x20);          /*Write EOI to 8259 Interrupt Controller*/
return;
```

Transmission

To transmit a string of data, the transmitter must first check Bit 5 of the Line Status Register at Base Address +5. That bit is the transmitter-holding-register-empty flag. If it is high, the transmitter has sent the data. The process of checking the bit until it goes high followed by a write is repeated until no data remains.

The following C code fragment demonstrates this process:

```
while(data[i]);    /*While there is data to send*/
{
 while(!(inportb(BASEADDR +5)&0x20));    /*Wait until transmitter is empty*/
 outportb(BASEADDR,data[i]);
 i++;
}
```

Chapter 6: Connector Pin Assignments

Input/Output Connections

The PCI-COM422/8 Serial Communications card uses a 50-pin SCSI D-connector to interface to a spider cable. The spider cable has eight individual 9-pin connectors provided with it.

To ensure that there is minimum susceptibility to EMI and minimum radiation it is important that the card mounting bracket be properly screwed into place and that there be a positive chassis ground. Also, proper EMI cabling techniques (cable connect to chassis ground at the aperture, shielded twisted-pair wiring, etc) must be used for the input/output wiring.

Pin Number	RS-422 Signals	Pin Number	RS-422 Signals
Pin 1	GND Ground	Pin 26	GND Ground
Pin 2	Aout+	Pin 27	Eout+
Pin 3	Aout-	Pin 28	Eout-
Pin 4	Ground through 100 Ohm R	Pin 29	Ground through 100 Ohm R
Pin 5	Ain+	Pin 30	Ein+
Pin 6	Ain-	Pin 31	Ein-
Pin 7	GND Ground	Pin 32	GND Ground
Pin 8	Bout+	Pin 33	Fout+
Pin 9	Bout-	Pin 34	Fout-
Pin 10	Ground through 100 Ohm R	Pin 35	Ground through 100 Ohm R
Pin 11	Bin+	Pin 36	Fin+
Pin 12	Bout-	Pin 37	Fout-
Pin 13	GND Ground	Pin 38	GND Ground
Pin 14	Cout+	Pin 39	Gout+
Pin 15	Cout-	Pin 40	Gout-
Pin 16	Ground through 100 Ohm R	Pin 41	Ground through 100 Ohm R
Pin 17	Cin+	Pin 42	Gin+
Pin 18	Cin-	Pin 43	Gin-
Pin 19	GND Ground	Pin 44	GND Ground
Pin 20	Dout+	Pin 45	Hout+
Pin 21	Dout-	Pin 46	Hout-
Pin 22	Ground through 100 Ohm R	Pin 47	Ground through 100 Ohm R
Pin 23	Din+	Pin 48	Hin+
Pin 24	Din-	Pin 49	Hin-
Pin 25	GND Ground	Pin 50	GND Ground

Table 6-1: Connection Pin Assignments

Signal	Connector Pin
Ain-	Pin 1
Aout+	Pin 2
Aout-	Pin 3
100 Ohm to Ground	Pin 5
Ain+	Pin 9

Table 6-2: Data Cable Wiring

Customer Comments

If you experience any problems with this manual or just want to give us some feedback, please email us at: ***manuals@accesioproducts.com.***. Please detail any errors you find and include your mailing address so that we can send you any manual updates.

ACCES I/O PRODUCTS, INC.
10623 Roselle Street, San Diego CA 92121
Tel. (619)550-9559 FAX (619)550-7322
www.accesioproducts.com